Service curves in Network Calculus: dos and don'ts

Anne Bouillard, Laurent Jouhet, Eric Thierry

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Service curves in Network Calculus: dos and don'ts

Anne Bouillard , Laurent Jouhet , Eric Thierry

N° 7094

Novembre 2009

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7094--FR+ENG

Service curves in Network Calculus: dos and don'ts

Anne Bouillard *, Laurent Jouhet †, Eric Thierry ‡

Thème : Réseaux et télécommunications
Équipe-Projet DistribCom

Rapport de recherche n° 7094 — Novembre 2009 — 24 pages

Abstract: This report is a survey about service curves used in envelope-based models for worst-case performance evaluation, like Network Calculus or Real-Time Calculus. In a first part, we compare different models and their notions of service curves by establishing a hierarchy and similarities between different types of service curves and in a second part, we compare the behaviours of simple and strict service curves for concatenation and residual service curves.

Key-words: Network Calculus, politique de service

* ENS Cachan (Brittany) / IRISA, Anne.Bouillard@bretagne.ens-cachan.fr
† ENS Lyon / LIP / IXXI, Laurent.Jouhet@ens-lyon.fr
‡ ENS Lyon / LIP / IXXI, Eric.Thierry@ens-lyon.fr

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu, 35042 Rennes Cedex
Téléphone : +33 2 99 84 71 00 — Télécopie : +33 2 99 84 71 71

Courbes de service en *Network Calculus*

Résumé : Ce rapport est un état de l'art sur les courbes de service utilisées pour l'analyse de performances dans le pire cas basé sur des modèles d'enveloppe, comme le *Network Calculus*. Dans un premier temps, nous comparons différents modèles et leurs modèles de courbes de service en établissant une hiérarchie entre eux et en mettant en évidence les similarités de ces modèles. Dans un second temps, nous comparons le comportement des courbes de service simple et des courbes de service strict pour la concaténation et le service résiduel.

Mots-clés : *Network Calculus*, politique de service

1 Introduction

Network Calculus is a theory of deterministic queuing systems encountered in communications networks. It is based on $(\min, +)$ algebra and it can be seen as a $(\min, +)$ filtering theory by analogy with the $(+, \times)$ filtering theory used in traditional system theory. More than just a formalism, it enables to analyze complex systems and to prove deterministic bounds on delays, backlogs and other Quality-of-Service (QoS) parameters. The analysis usually focuses on worst-case performances. The information about the system features are stored in functions, such as arrival curves shaping the traffic or service curves quantifying the service guaranteed at the network nodes. These functions can be combined together thanks to special Network Calculus operations, in order to analyze the system and compute bounds on local performances (*i.e.* maximum buffer size at a node) or on end-to-end performances (*i.e.* maximum end-to-end delay. At the present time, the theory has developed and yield accomplished results which are mainly recorded in two reference books: Chang's book [6] and Le Boudec and Thiran's book [12].

Nevertheless it remains difficult to draw the exact borders of Network Calculus at the time being. One of the main obstacles comes from the apparent variety of *service curve* definitions in the literature which might lead to different types of models. In some papers, the use of one definition instead of another one does not seem to play an important role, whereas some other papers cautiously warn the readers to use the right definition to ensure relevancy or validity. One objective of this report is to unveil the differences between models yielded by the different definitions. Some of the results presented here could be classified as folklore theorems since they are sometimes quoted in the literature without proofs. We try to fill those gaps by proposing a proof for each of our statements.

Even if Network Calculus is still a developing theory without clear frontiers, some people have already described alternate theories using nevertheless close formalisms (envelope-based models, $(\min, +)$ operators) but claimed to be more precise or more relevant to their own applications. One can cite Real-Time Calculus, Sensor Calculus... A small presentation with apparent similarities/differences is presented in Table 1. The *Applications* line states the applications originally intended, or said differently the community that was involved in the first communications.

	Network Calculus (NC)	**Real-time Calculus (RTC)**	**Sensor Calculus (SC)**
Math.	envelope-based constraints, $(\min, +)$ algebra, univariate cumulative functions	envelope-based constraints, $(\min, +)$ algebra, bivariate cumulative functions	
App.	communication networks, Internet IntServ & DiffServ	real-time embedded systems	sensor networks
Soft.	DISCO Network Calculator [15, 9], COINC [2], Rockwell Collins ConfGen	RTC Toolbox [19, 17, 20, 5], CyNC [13, 14]	

Table 1: Comparison between several theories based on the (min,plus) algebra.

One can question whether the models analyzed in those theories are really different from Network Calculus models. If so, where do the differences lie? and are they deep? We will try to give partial answers to those questions, in particular concerning Real Time Calculus. We will see that rather than alternate theories we are in presence of extensions of Network Calculus usual models (*e.g.* by taking into account *maximal services* and not only *minimal services*).

Note that we stay focused on models without probabilistic assumptions and theories aiming at computing (deterministic) worst case performances. On the probabilistic side, several formalisms have been presented in the literature under the banner "Stochastic Network Calculus" [6, 7, 8, 10, 11]. They still use some envelope-based constraints, but with probabilistic assumptions, and still aim at evaluating worst case performances which now follow a random distributions. We will not deal with such probabilistic models in this report, although this field of investigation seems very attractive for future researches.

2 Comparison of current notions of service curves

2.1 Definitions and notation

NC functions and operations. Network Calculus primal objective is the performance analysis of communication networks. Flows and services in the network are modelled by non-decreasing functions $t \mapsto f(t)$ where t is *time* and $f(t)$ an amount of *data*. There are different models depending on whether t (resp. $f(t)$) takes discrete or continuous values, *e.g.* in $\mathbb{N}$ or $\mathbb{R}_+$. Concerning data, one can even consider three models of granularity:

- ***infinitesimal*** (sometimes called ***fluid***): data can be divided into arbitrarily small pieces (data measures are in $\mathbb{R}_+$);
- ***unitary*** (sometimes called ***discrete***): data is composed of indivisible packets of the same length (data measures are in $\mathbb{N}$),
- ***multi-scaled***: data may be divided into packets of variable lengths (possibly in $\mathbb{N}$, $\mathbb{R}_+$ or even infinitesimal).

In all cases, we will use the term *bit* as the data unit of measure. Note also that we do not exactly use the term *fluid* as in [12] where the *fluid model* adds the condition that the manipulated functions are continuous.

In Network Calculus, one must distinguish two kinds of objects: the real movements of data and the constraints that these movements satisfy. The real movements of data are mainly modeled by ***cumulative functions***: a cumulative function $f(t)$ counts the total amount of data that has achieved some condition up to time t (*e.g.* the total amount of data which has gone through a given place in the network). In all the paper, ***we make the usual assumption that cumulative functions are left-continuous***. This is not a huge restriction for the modeler: *e.g.* let $f(t)$ be the total amount of data that has entered a system until time t, in case an instantaneous burst of b bits occur at time t_0 while a bits had already arrived, one only has to set $f(t_0) = a$ and $f(t_0+) = a + b$. This assumption has nevertheless a technical importance in the Network Calculus edifice (*e.g.* when defining the start of backlogged periods and using strict service curves as in Theorem 5). On the contrary, no assumption of (left- or right-)continuity is imposed to the constraint functions.

Network Calculus functions belong to $\mathcal{F}$ (resp. $\mathcal{D}$) the set of functions (non necessarily non-decreasing) from $\mathbb{R}_+$ (resp. $\mathbb{N}$) into $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$, if time is continuous (resp. discrete). Cumulative functions usually belong to $\mathcal{F}_\uparrow = \{f \in \mathcal{F} \mid f \text{ non-decreasing, left-continuous, } f(0) = 0\}$ (resp. $\mathcal{D}_\uparrow = \{f \in \mathcal{D} \mid f \text{ non-decreasing , } f(0) = 0\}$)

Beyond usual operations like the minimum or the addition of functions, Network Calculus makes use of several classical operations [1] which are the translations of $(+, \times)$ filtering operations into the $(\min, +)$ setting, as well as a few other transformations. Here is a sample of operations that can be encountered: let $f, g \in \mathcal{F}$ (resp. $\mathcal{D}$), $\forall t \in \mathbb{R}_+$ (resp. $\mathbb{N}$),

- (Inf-)convolution: $(f * g)(t) = \inf_{0 \le s \le t}(f(s) + g(t-s))$.
- Sup-convolution: $(f \overline{*} g)(t) = \sup_{0 \le s \le t}(f(s) + g(t-s))$.
- (Sup-)deconvolution: $(f \oslash g)(t) = \sup_{u \ge 0}(f(t+u) - g(u))$.
- Inf-deconvolution:$(f \overline{\oslash} g)(t) = \inf_{u \ge 0}(f(t+u) - g(u))$.
- Positive rounding: $f_+(t) = \max(f(t), 0)$.
- Positive and non-decreasing upper closure: $f_\uparrow(t) = \max(\sup_{0 \le s \le t} f(s), 0)$.

Such operations have interesting algebraic properties (see [1, 12] for surveys). Using those operations, Network Calculus formulas combine the constraints on the traffic and the services in the network in order to output worst-case performance bounds.

For now on, we will do the presentation with *continuous time* and thus with functions in $\mathcal{F}$. But one will easily check that all the results will remain true with *discrete time* and functions in $\mathcal{D}$.

NC input/output systems. An NC model for a communication network usually consists in:

- a partition of the network into subsystems (often called *nodes*) which may have different scales (from elementary hardware like a processor to large sub-networks).
- a description of data flows, where each flow follows a path through a specified sequence of subsystems and where each flow is shaped by some arrival curve just before entering the network.
- a description of the behavior of each subsystem, that is service curves bounding the performances of each subsystem, as well as service policies in case of multiplexing (several flows entering the same subsystem and thus sharing its service).

Systems or sub-systems are described as input/output systems (where the number of inputs is the same as the number of outputs). An ***(acceptable) trajectory*** for a system crossed by p flows is a set of cumulative functions $(A_k)_{1\leq k\leq p}$ and $(B_k)_{1\leq k\leq p}$ in $\mathcal{F}_\uparrow$ (where A_k and B_k respectively correspond to the cumulative functions of flow k at the input and the output of the system). For now, a ***system*** $\mathcal{S}$ over p flows will be simply defined as the set of all its acceptable trajectories, that is $\mathcal{S} \subseteq \mathcal{F}_\uparrow^p \times \mathcal{F}_\uparrow^p$. Such a black boxed view is usual in classical filtering theory and enables to deal with any scale of system. Note also that this definition allows to consider *deterministic dynamics* (one output for one input) and *non-deterministic dynamics* (several possible outputs for one input).

NC main performance measures: backlog & delay. Let (A, B) be an input/output trajectory for a flow in a system. Then the *global backlog* of the flow at time t is $b(t) = A(t)-B(t)$ and the delay (under the FIFO policy assumption) endured after z input bits is $d(z) = B^{(-1)}(z)-A^{(-1)}(z)$ where for all $f \in \mathcal{F}$, $f^{(-1)}(z) = \inf\{t \geq 0 \mid f(t) \geq z\}$ (*pseudo-inverse*).

Let $\mathcal{S}$ be a system. The *worst-case backlog over* $\mathcal{S}$ is $b_{\max} = \sup_{(A,B)\in\mathcal{S}} \sup_{t\geq 0} A(t) - B(t)$. The *worst-case delay over* $\mathcal{S}$ is $d_{\max} = \sup_{(A,B)\in\mathcal{S}} \sup_{z\geq 0} B^{(-1)}(z) - A^{(-1)}(z)$.

Given a trajectory $(A, B) \in \mathcal{F}_\uparrow \times \mathcal{F}_\uparrow$, a ***backlogged period*** is an interval $I \subseteq \mathbb{R}_+$ of time during which the backlog is non-null, *i.e.* $\forall u \in I$, $A(u) - B(u) > 0$. Let $t \in \mathbb{R}_+$, the ***start of the backlogged period*** of t is $start(t) = \sup\{u \leq t | A(u) = B(u)\}$. Since the cumulative functions A and B are assumed left-continuous, we also have $A(start(t)) = B(start(t))$. If $A(t) = B(t)$, then $start(t) = t$. For any $t \in \mathbb{R}_+$, $]start(t), t[$ is a backlogged period ($]start(t), t]$ if $A(t) - B(t) > 0$).

In the definition of backlogged period, the interval I can be closed, semi-closed or open. Such a flexible definition is convenient in some future definitions or proofs where the precise description of trajectories requires a particular type of intervals, *e.g.* semi-closed rather than open (see the consequences of such choices in the Complement about strict service curves below). Note that in the literature, backlogged periods have been sometimes defined for open intervals only [4] (page 885) or without worrying about this question [12] (Definition 1.3.2, page 21).

NC arrival curves: one definition. Given a data flow traversing a network, let $A \in \mathcal{F}_\uparrow$ be its *cumulative function* at some point in the network, *i.e.* $A(t)$ is the number of bits that have gone through this point until time t, with $A(0) = 0$. A function $\alpha \in \mathcal{F}$ is an ***arrival curve*** for A if $\forall\ s, t \in \mathbb{R}_+$, $0 \leq s \leq t$, we have $A(t) - A(s) \leq \alpha(t - s)$.

The set of all arrival curves for $A \in \mathcal{F}_\uparrow$ admits a minimum which remains an arrival curve: it is $\alpha = A \oslash A$ and we call it the *canonical arrival curve* for A.

NC service curves: several definitions. In the literature, the definitions of service curves usually concern:

- *minimum service curves* which are lower bounds on the service provided in a system (useful for upper bounds on worst case performances).
- *single flow systems* $\mathcal{S}$, that is $\mathcal{S} \subseteq \mathcal{F}_{\uparrow} \times \mathcal{F}_{\uparrow}$.

Note that in NC models with multiplexing, the aggregation (Σ) of all the flows entering the system is often considered as a single flow to which the minimum service is applied.

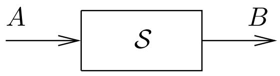


Figure 1: A single flow input/output system.

For each type $\mathcal{T}$ of service curve, we define for any $\beta \in \mathcal{F}$ and for any input/output trajectory $(A, B) \in \mathcal{F}_{\uparrow} \times \mathcal{F}_{\uparrow}$ the conditions so that β *is a* $\mathcal{T}$*-service curve for* (A, B) (we also say that (A, B) *admits* β *as a* $\mathcal{T}$*-service curve*). We then define for all $\beta \in \mathcal{F}$, $\mathcal{S}_{\mathcal{T}}(\beta)$ the set of all trajectories admitting β as a $\mathcal{T}$-service curve. We say that a system $\mathcal{S}$ admits β as a $\mathcal{T}$-service curve if it is true for all its trajectories, *i.e.* $\mathcal{S} \subseteq \mathcal{S}_{\mathcal{T}}(\beta)$.

- ***Simple service curve:*** $\mathcal{S}_{simple}(\beta) = \{(A, B) \in \mathcal{F}_{\uparrow} \times \mathcal{F}_{\uparrow} \mid A \geq B \geq A * \beta\}$.
- ***Strict service curve (weak sense):*** $\mathcal{S}_{wstrict}(\beta) = \{(A, B) \in \mathcal{F}_{\uparrow} \times \mathcal{F}_{\uparrow} \mid A \geq B, \text{ and } \forall t \geq 0,\ B(t) - B(start(t)) \geq \beta(t - start(t))\}$.
- ***Strict service curve:***
$\mathcal{S}_{strict}(\beta) = \{(A, B) \in \mathcal{F}_{\uparrow} \times \mathcal{F}_{\uparrow} \mid A \geq B, \text{ and } \forall \text{ backlogged period }]s, t], B(t) - B(s) \geq \beta(t - s)\}$.
- ***Variable capacity node:*** $\mathcal{S}_{vcn}(\beta) = \{(A, B) \in \mathcal{F}_{\uparrow} \times \mathcal{F}_{\uparrow} \mid \exists C \in \mathcal{F}_{\uparrow}, \forall t \geq 0,\ B(t) = \inf_{0 \leq s \leq t} \left[A(s) + C(t) - C(s)\right] \text{ and } \forall 0 \leq s \leq t,\ C(t) - C(s) \geq \beta(t - s)\}$.

Two classical functions used as service curves are:

- *Pure delay* $T \in \mathbb{R}_+ \cup \{+\infty\}$: $\delta_T(t) = 0$ if $t \leq T$, $= +\infty$ otherwise.
- *Constant rate* $r \in \mathbb{R}_+ \cup \{+\infty\}$: $\lambda_r(t) = rt$ (if $r = +\infty$, we set $\lambda_r = \delta_0$).

Nothing prevents from using some service curves which are not in $\mathcal{F}_{\uparrow}$, *e.g.* with negative values, decreasing parts or left-discontinuities. Nevertheless note that it is usually required that at least $\beta(0) \leq 0$, otherwise $\beta(0) > 0$ require that $B(0) > A(0)$ and $\mathcal{S}_{simple}(\beta) = \mathcal{S}_{wstrict}(\beta) = \mathcal{S}_{strict}(\beta) = \mathcal{S}_{vcn}(\beta) = \emptyset$.

Complement about strict service curves. The definition of strict service curves presented here is the one used by Schmitt et al [15, 16]. Some papers do not choose exactly the same definition for strict service curves [3, 4]: they replace the backlogged interval $]s, t]$ in the definition by $]s, t[$ (both definitions allow $B(s) = A(s)$, but this variant also allows $B(t) = A(t)$). For $\beta \in \mathcal{F}$, let us denote $\mathcal{S}'_{strict}(\beta)$ the set of trajectories satisfying this variant of our definition. How do those slightly different definitions compare ? It is clear that $\forall \beta \in \mathcal{F}$, $\mathcal{S}'_{strict}(\beta) \subseteq \mathcal{S}_{strict}(\beta)$. If β is left-continuous, since all cumulative functions are assumed to be left-continuous, we have the equality $\mathcal{S}'_{strict}(\beta) = \mathcal{S}_{strict}(\beta)$: let $(A, B) \in \mathcal{S}_{strict}(\beta)$, if $]s, t[$ is a backlogged period, then $\forall s < t' < t$, $]s, t']$ is a backlogged period, thus $B(t') - B(s) \geq \beta(t' - s)$ and $B(t) - B(s) \geq \beta(t - s)$ by letting t' tend to t. If β is not left-continuous, then we may have a strict inclusion $\mathcal{S}'_{strict}(\beta) \subsetneq \mathcal{S}_{strict}(\beta)$. Consider the example of Figure 2 where $A(t) = 1/2$ if $t > 0$ and $= 0$ if $t = 0$, $B(t) = \min(t/2, 1/2)$ and $\beta(t) = \lfloor t \rfloor$. There is a unique maximal backlogged period which is $]0, 1[$. One can check that

$(A,B) \in \mathcal{S}_{strict}(\beta)$ (since $\forall]s,t]$, $0 \leq s \leq t < 1$, $B(t) - B(s) \geq \beta(t-s) = 0$), but $(A,B) \notin \mathcal{S}'_{strict}(\beta)$ (since $B(1) - B(0) = 1/2 \not\geq \beta(1-0) = 1$). Note by the way that $(A,B) \in \mathcal{S}_{vcn}(\beta)$ (*e.g.* one can choose $C(t) = B(t)$ if $t < 1$ and $= t$ if $t \geq 1$). It is not surprising since Theorem 1 will show that $\mathcal{S}_{vcn}(\beta) \subseteq \mathcal{S}_{strict}(\beta)$.

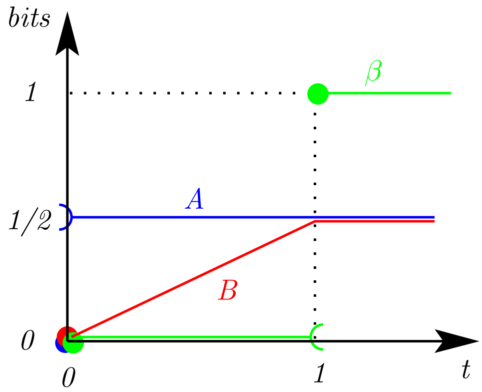


Figure 2: Beware of the definition of strict service curves.

Complement about Variable Capacity Nodes.

Lemma 1 (Consistence and alternate characterization)**.** *Let $A, C \in \mathcal{F}_\uparrow$ and $B \in \mathcal{F}$ such that $\forall t \geq 0$, $B(t) = \inf_{0 \leq s \leq t} \left[A(s) + C(t) - C(s)\right]$. Then $B \in \mathcal{F}_\uparrow$ and $\forall t \geq 0$, $B(t) = A(start(t)) + C(t) - C(start(t))$.*

Proof. Since $A(0) = C(0) = 0$, it is clear that $B(0) = 0$. Note that by hypothesis, $\forall t \geq 0$, $B(t) \leq A(t)$ (choose $s = t$) and $B(t) \leq C(t)$ (choose $s = 0$).

Since A and C are left-continuous, one can check that the formula for B ensures that it is also left-continuous.

Let $t < t'$, since A and C are non-decreasing, we have:

$$B(t') = \inf \Big[\underbrace{\inf_{0 \leq s \leq t} \left(A(s) + C(t') - C(s)\right)}_{=C(t')-C(t)+B(t) \geq B(t)}, \underbrace{\inf_{t \leq s \leq t'} \left(A(s) + C(t') - C(s)\right)}_{=A(t)+C(t')-C(t') \geq B(t)} \Big].$$

Thus $B(t') \geq B(t)$, that is B is non-decreasing.

For all $t \geq 0$, $start(t)$ is well defined due to left-continuity of A and B and $A(0) = B(0) = 0$. For all $t \geq 0$, we have:

$$B(t) = \inf \Big[\underbrace{\inf_{0 \leq s \leq start(t)} \left[A(s) + C(t) - C(s)\right]}_{\substack{= B(start(t)) + C(t) - C(start(t)) \\ = A(start(t)) + C(t) - C(start(t))}}, \inf_{start(t) \leq s \leq t} \left[A(s) + C(t) - C(s)\right] \Big].$$

The first term $A(start(t)) + C(t) - C(start(t))$ appears in the second term when taking $s = start(t)$. Thus we have for all $t \geq 0$,

$$(VCN) \qquad B(t) = \inf_{start(t) \leq s \leq t} \left[A(s) + C(t) - C(s)\right].$$

Now if $B(t) = A(t)$, we have $start(t) = t$ and thus $A(start(t)) + C(t) - C(start(t)) = A(t) = B(t)$ which concludes the proof. Otherwise $B(t) < A(t)$ (and thus $start(t) < t$). In this case,

$$\forall start(t) < s \leq t, \; A(s) > B'(s) = \inf_{start(s) \leq u \leq s} \left[A(u) + C(s) - C(u)\right],$$

with $start(s) = start(t)$ by definition. Thus $\forall start(t) < s \leq t$,

$$A(s)+C(t)-C(s) > \inf_{start(t)\leq u\leq s} \big[A(u)+C(t)-C(u)\big] \geq \inf_{start(t)\leq u\leq t} \big[A(u)+C(t)-C(u)\big] = B(t)$$

Thus the infimum in Equation (VCN) is necessarily reached at $s = start(t)$, that is

$$B(t) = A(start(t)) + C(t) - C(start(t)).$$

Note that to prove this formula we have not used the non-decreasing property of A, B, C. □

2.2 Comparison of NC service curves

All the definitions from the literature share the same natural monotonic behavior about trajectories.

Proposition 1 (Monotony). *For any type $\mathcal{T}$ of service curve in the literature, for all $\beta, \beta' \in \mathcal{F}$ (not necessarily in $\mathcal{F}_\uparrow$), if $\beta \leq \beta'$ then $\mathcal{S}_\mathcal{T}(\beta) \supseteq \mathcal{S}_\mathcal{T}(\beta')$.*

Moreover, for strict and weakly strict service curves, one can replace service curves by their closure.

Proposition 2. *Let $\beta \in \mathcal{F}$, then $\mathcal{S}_{wstrict}(\beta) = \mathcal{S}_{wstrict}(\beta_\uparrow)$, $\mathcal{S}_{strict}(\beta) = \mathcal{S}_{strict}(\beta_\uparrow)$ and $\mathcal{S}_{strict}(\beta) = \mathcal{S}_{strict}(\overline{\beta^*})$.*

A hierarchy can now be established between all those notions of service curves. The hierarchy given below is strict in most of the cases (the inclusions are strict). Nevertheless, this is not always true.

Theorem 1 (Hierarchy). *For all $\beta \in \mathcal{F}$, we have the following inclusions:*

$$\mathcal{S}_{vcn}(\beta) \subseteq \mathcal{S}_{strict}(\beta) \subseteq \mathcal{S}_{wstrict}(\beta) \subseteq \mathcal{S}_{simple}(\beta).$$

Proof. We suppose that $\beta(0) \leq 0$, otherwise all the sets are empty.

The inclusion $\mathcal{S}_{strict}(\beta) \subseteq \mathcal{S}_{wstrict}(\beta)$ is clear, since for all $t \in \mathbb{R}_+$, either $]start(t), t]$ is a backlogged period and thus $B(t) - B(start(t)) \geq \beta(t - start(t))$, or $]start(t), t[$ is a backlogged period and $B(t) = A(t)$ but then $start(t) = t$ and $B(t) - B(start(t)) = 0 \geq \beta(t - start(t)) = 0$.

The inclusion $\mathcal{S}_{wstrict}(\beta) \subseteq \mathcal{S}_{simple}(\beta)$ comes from the remark that if $(A, B) \in \mathcal{S}_{wstrict}(\beta)$, then $B(t) \geq B(start(t)) + \beta(t - start(t)) = A(start(t)) + \beta(t - start(t)) \geq \inf_{0\leq s\leq t}(A(s) + \beta(t - s))$.

Now let us show that $\mathcal{S}_{vcn}(\beta) \subseteq \mathcal{S}_{strict}(\beta)$. Let $(A, B) \in \mathcal{S}_{vcn}(\beta)$, there exists $C \in \mathcal{F}_\uparrow$ such that $\forall t \geq 0$, $B(t) = \inf_{0\leq s\leq t} \big[A(s) + C(t) - C(s)\big]$ and $\forall 0 \leq s \leq t$, $C(t) - C(s) \geq \beta(t - s)$. Consider a backlogged period $]s, t]$ for (A, B), then by definition $start(s) = start(t) = p$. From Lemma 1, we have $B(t) = B(p)+C(t)-C(p)$ and $B(s) = B(p)+C(s)-C(p)$. Thus $B(t)-B(s) = C(t) - C(s) \geq \beta(t - s)$ and we have proved that β is a strict service curve for (A, B). □

3 RTC vs NC

Real Time Calculus is presented as an alternative to Network Calculus. Admitting some kinship like the use of envelopes shaping the cumulative curves of traffic and services, it also claims new features allowing more precise models, tighter bounds and a way to compute them in any complex system.

In the literature, the comparisons with NC are very informal [20]. We now try to provide a mathematical comparison between RTC models and NC models. The RTC presentation is extracted from the very good and comprehensive Wandeler's PhD Thesis [19], in particular its Appendix A, page 197 (proofs and details are often omitted in other references). All the RTC notation and results below are extracted from this thesis.

To be precise, we will draw the comparison between *RTC Greedy Processors (GP)* and *NC Variable Capacity Nodes.*

RTC Greedy Processor	NC Variable Capacity Node
$R[s,t)$, $s \leq t$, $s,t \in \mathbb{R}$, non-negative, satisfying the Chasles relation: $\forall u \leq v \leq w$, $R[u,w) = R[u,v) + R[v,w)$.	$R(t)$, $t \in \mathbb{R}$ or more often $t \in \mathbb{R}_+$, non-decreasing and $R(0) = 0$.

3.1 Spaces of functions.

Here are the main differences which are listed in RTC references [20]:

- RTC temporal variables range over all $\mathbb{R}$ (and not just $\mathbb{R}_+$ as it is often the case in NC),
- RTC gets rid of the NC *initialization point* $t = 0$ where all NC functions are null,
- RTC functions have two arguments instead of one in NC (which is presented as one way to deal with all $\mathbb{R}$).

Here is one important omission in the hypotheses about RTC functions:

- the Chasles relation is never explicitly exposed in the hypotheses, although it is intensively used in the proofs. Of course the interpretation of $R[s,t)$ as the amount of data (or service) provided during the interval $[s,t[$ suggests the Chasles relation. Nevertheless a formal presentation should mention the relation.

Now what about the claims that RTC functions have a better modeling power with tighter bounds [19, 20]? They are rather unclear (due to the Chasles relation) as explained below.

Let $R(.)$ a function defined on $\mathbb{R}$ (if it is defined on $\mathbb{R}_+$, an usual and natural extension is $R(t) = 0$ if $t < 0$). Then we define $\hat{R}[s,t) = R(t) - R(s)$ for all s,t. One can check that $\hat{R}$ satisfies the Chasles relation and if R is non-decreasing, then $\hat{R}$ is non-negative.

Let $R[.,.)$ a function defined on $\{(s,t) \in \mathbb{R}^2 \mid s \leq t\}$. Then we define $\check{R} = R[0,t)$ for $t \geq 0$ and $= -R[t,0)$ for $t < 0$. If R satisfies the Chasles relation, then it is easy to check that $\check{R}(0) = 0$ and $\check{R}(t) - \check{R}(s) = R[s,t)$ for all $s \leq t$ (*i.e.* $\hat{\check{R}} = R$). Thus if R is non-negative, then $\check{R}$ is non-decreasing (note that all of this also works for $\check{R} = R[p,t)$ with an arbitrary $p \in \mathbb{R}$, except that $\check{R}(0)$ is not necessarily 0).

We sum up these simple remarks in the next proposition.

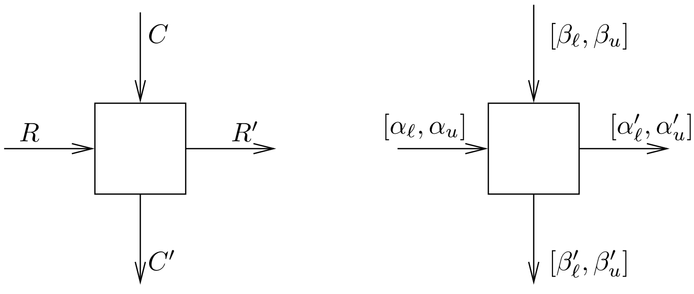
Proposition 3. *The mapping $R \mapsto \check{R}$ is a bijection between non-negative functions over $\{(s,t) \in \mathbb{R}^2 \mid s \leq t\}$ with the Chasles relation and non-decreasing functions over $\mathbb{R}$ with null value at $t = 0$. Its inverse is the mapping $R \mapsto \hat{R}$.*

Thus the only justification for the use of RTC functions with two arguments might be that it is easier to write the equations for the dynamics.

3.2 RTC equations

RTC Greedy Processor	NC Variable Capacity Node
Time: $t \in \mathbb{R}$ Input functions: R, C Output functions: R', C' Backlog: b	Time: $t \in \mathbb{R}_+$ Input functions: R, C Output functions: R', C' Backlog: b
$(A7)$ $R'[s,t) = C[s,t) - C'[s,t)$ $(A8)$ $C'[s,t) = \sup_{s \leq u \leq t} \big[C[s,u) - R(s,u) - B(s), 0\big]$ $(A10)$ $b(t) - b(s) = R[s,t) - R'[s,t)$	$(B1)$ $R'(t) = \inf_{0 \leq u \leq t} \big[C(t) - C(u) + R(u)\big]$ $(B2)$ $C'(t) = C(t) - R'(t)$ $(B3)$ $b(t) = R(t) - R'(t)$

When looking at Variable Capacity Node equations, one can easily see that for *any* input functions R and C, there exists output functions R', C' satisfying the equations and they are



(a) (b)

Figure 3: A single flow input/output system. (a) cumulative functions; (b) arrival/service lower and upper arrival curves.

unique (($B1$) gives R' from R and C, then ($B2$) gives C' from R' and C). The backlog is also well defined in an unique way (with ($B3$)). When looking at RTC, given some input functions R and C, the existence of functions R', C', b satisfying the equations is not obvious (two unknown functions per equation).

Lemma 2 (Coherence of RTC equations)**.** *Given R, C, let R', C', b be some solutions (if any) to the system of equations ($A7$), ($A8$), ($A10$). If R and C satisfy the Chasles relation and are non-negative, then so do R' and C'. In this case, if R and C are also non-negative, then so are B, R', C'.*

Proof. Concerning the preservation of the Chasles relation, due to Equation ($A7$) and since C satisfies the Chasles relation, it is sufficient to prove that either R' or C' satisfies the relation so that both of them do.

Let $s \leq u \leq t$, we have:

$$\begin{aligned}
C'[s,u] + C'[u,t] &\overset{(A8)}{=} C'[s,u] + \sup_{u \leq w \leq t} \Big[\underbrace{C[u,w]}_{C[s,w]-C[s,u]} - \underbrace{R[u,w]}_{R[s,w]-R[s,u]} - b(u), 0 \Big] \\
&\overset{(A10)}{=} C'[s,u] + \sup_{u \leq w \leq t} \big[C[s,w] - R[s,w] + R'[s,u] - C[s,u] - b(s), 0 \big] \\
&\overset{(A7)}{=} C'[s,u] + \sup_{u \leq w \leq t} \big[C[s,w] - R[s,w] - C'[s,u] - b(s), 0 \big] \\
&= \sup_{u \leq w \leq t} \big[C[s,w] - R[s,w] - b(s), C'[s,u] \big] \\
&= \sup_{u \leq w \leq t} \big[C[s,w] - R[s,w] - b(s), \sup_{s \leq w \leq u} [C[s,w] - R[s,w] - b(s), 0] \big] \\
&= \sup_{s \leq w \leq t} \big[C[s,w] - R[s,w] - b(s), 0 \big] \\
&= C'[s,t].
\end{aligned}$$

Now, if C' satisfies the Chasles relation, then for all $s \in \mathbb{R}$, $C'[s,s] = 0$. That is $\sup \big[C[s,s] - R[s,s] - b(s), 0 \big] = 0$ and thus $b(s) \geq 0$. Moreover, since C' is a supremum with 0, $C'[s,t] \geq 0$ for all $s \leq t$. Then checking whether $R'[s,t] \geq 0$ comes to checking that $C'[s,t] \leq C[s,t]$. This is true because $C'[s,t] = \sup_{s \leq u \leq t} \big[\underbrace{C[s,u]}_{\leq C[s,t]} - \underbrace{R[s,u]}_{\geq 0} - \underbrace{b(s)}_{\geq 0}, \underbrace{0}_{\leq C[s,t]} \big]$. □

Lemma 3 (A recursive equation)**.** *Let $A \in \overline{\mathbb{R}}^{\mathbb{R}}$, then the functions $F \in \overline{\mathbb{R}}^{\mathbb{R}}$ satisfying:*

$$\forall s \leq t, \; F(t) = \inf \big(\inf_{s \leq u \leq t} A(u), F(s) \big)$$

are exactly the functions of the form $F(t) = \inf\left(\inf_{-\infty \le u \le t} A(u), a\right)$ *where* $a \in \overline{\mathbb{R}}$ *is a constant.*

Proof. We proceed by necessary conditions. Any function $F \in \overline{\mathbb{R}}^{\mathbb{R}}$ satisfying the initial relation are such that:

- $\forall s \le t,\ F(t) \le F(s)$, thus F is non-increasing over $\mathbb{R}$. It admits a limit $\lim_{s \to -\infty} F(s) = a \in \overline{\mathbb{R}}$.
- $\forall s \le t,\ F(t) \le \inf_{s \le u \le t} A(u)$. Thus $F(t) \le \inf_{-\infty \le u \le t} A(u)$. It also means that we can write $F(t) = \inf\left(\inf_{-\infty \le u \le t} A(u), F(s)\right)$

This last expression for F is true for all $s \le t$. Take the limit $s \to -\infty$ and we obtain $F(t) = \inf\left(\inf_{-\infty \le u \le t} A(u), a\right)$.

Conversely, it is easy to check that any function of this type satisfies the initial relation. □

Theorem 2 (Equivalence RTC - NC)**.** *Let R and C be two functions defined over $D = \{(s,t) \in \mathbb{R}^2 \mid s \le t\}$, non-negative and satisfying the Chasles relation. Let $p \in \mathbb{R}$ be an arbitrary real number and for any function F defined over D, let $\check{F}(t) = F[p,t)$ if $t \ge p$, and $= -F[t,p)$ if $t < p$. Let $\omega = \inf_{-\infty \le u \le p}\left[\check{R}(u) - \check{C}(u)\right]$.*

The set of solutions R', C', b to the system of equations $(A7)$, $(A8)$, $(A10)$ is given by $R'[s,t) = \check{R}'(t) - \check{R}'(s)$, $s \le t$ (and the same for C') and for all $t \in \mathbb{R}$,

$$\begin{aligned}
\check{R}'(t) &= \check{C}(t) + \inf\Big(\inf_{-\infty \le u \le t}\left[\check{R}(u) - \check{C}(u) + \sigma\right], \gamma\Big)\\
\check{C}'(t) &= \check{C}(t) - \check{R}'(t)\\
b(t) &= \check{R}(t) - \check{R}'(t) + \sigma,
\end{aligned}$$

where $\sigma, \gamma \in \overline{\mathbb{R}}$ are constants such that ($\sigma = -\omega$ and $\gamma \ge 0$), or ($\sigma \ge -\omega$ and $\gamma = 0$).

We always have $b(p) = \sigma$, and in case $b(p) = 0$, then for all $t \ge p$,

$$\check{R}'(t) = \inf_{p \le u \le t}\left[\check{R}(u) + \check{C}(t) - \check{C}(u)\right].$$

which is a Variable Capacity Node description with initial point at $t = p$ (both functions $\check{R}$ and $\check{C}$ are non-decreasing, equal to 0 at $t = p$, and non-negative over $[p, +\infty[$).

Proof. We proceed by necessary conditions to find some closed form expressions of the solutions of the system $(A7)$, $(A8)$, $(A10)$.

Under our hypotheses about R and C, due to Lemma 2, R' and C' must satisfy the Chasles relation. *Fix $p \in \mathbb{R}$ some arbitrary real number.* Following Proposition 3 and using the notation $\check{F}(t) = F[p,t)$ if $t \ge p$ and $= -F[p,t)$ if $t < p$, for any function $F[.,.)$ defined over $\{(s,t) \in \mathbb{R}^2 \mid s \le t\}$, we know that R' (resp. C') will be fully defined by the values of $\check{R}'$ (resp. $\check{C}'$). We can rewrite and mix Equations $(A7)$, $(A8)$, $(A10)$ where the unknowns are now $\check{R}', \check{C}' \in \mathcal{F}_{\uparrow}$ and $b \in \mathcal{F}$. Here is a set of equations they must satisfy.

First $\check{R}'(p) = 0$ and $\check{C}'(p) = 0$. When the Chasles relation is satisfied everywhere, $(A7)$ is clearly equivalent to $(A7')$: $\forall t \in \mathbb{R}$, $\check{R}'(t) = \check{C}(t) - \check{C}'(t)$. Since we know that $\check{C}(p) = 0$, $\check{R}'(p) = 0$ implies $\check{C}'(p) = 0$ and reciprocally.

Mixing $(A7)$ and $(A8)$ gives: for all $s \le t$,

$$R'[s,t) = C[s,t) - \sup_{s \le u \le t}\left[C[s,u) - R[s,u) - b(s), 0\right].$$

With our one-argument functions, it gives: for all $s \le t$,

$$\check{R}'(t) = \inf_{s \le u \le t}\left[\check{C}(t) - \check{C}(u) + \check{R}(u) - \check{R}(s) + b(s) + \check{R}'(s), \check{C}(t) - \check{C}(s) + \check{R}'(s)\right].$$

Equation $(A10)$ implies in particular that for all $s \geq p$, $b(s) - b(p) = R[p,s) - R'[p,s)$, that is $(A10')$: $b(s) - b(p) = \check{R}(s) - \check{R}'(s)$. When $s < p$, $b(p) - b(s) = R[s,p) - R'[s,p) = -\check{R}(s) + \check{R}'(s)$. Thus $(A10')$ is satisfied for any $s \in \mathbb{R}$.

The expression $\check{R}'(t)$ is thus equal to:

$$\check{R}'(t) = \inf_{s \leq u \leq t} \big[\check{C}(t) - \check{C}(u) + \check{R}(u) + b(p), \check{C}(t) - \check{C}(s) + \check{R}'(s)\big].$$

It can be reformulated into $(A8')$: for all $s \leq t$,

$$\check{R}'(t) - \check{C}(t) = \inf\big(\inf_{s \leq u \leq t} \big[\check{R}(u) - \check{C}(u) + b(p)\big], \check{R}'(s) - \check{C}(s)\big).$$

Applying Lemma 3, the function $\check{R}'$ is of the form: for all $t \in \mathbb{R}$,

$$\check{R}'(t) - \check{C}(t) = \inf\big(\inf_{-\infty \leq u \leq t} \big[\check{R}(u) - \check{C}(u) + b(p)\big], \gamma\big),$$

where $\gamma \in \overline{\mathbb{R}}$ is a constant.

The new system of Equations $(A7')$, $(A8')$, $(A10')$ and $\check{R}'(p) = 0$ has clearly some solutions $\check{R}'$, $\check{C}'$, b: one has to adjust $b(p)$ and γ so that $\check{R}'(p) = 0$, *i.e.* $\inf\big(\inf_{-\infty \leq u \leq p} \big[\check{R}(u) - \check{C}(u) + b(p)\big], \gamma\big) = 0$. Let us denote $\omega = \inf_{-\infty \leq u \leq p} \big[\check{R}(u) - \check{C}(u)\big]$. We know that $\omega \leq 0$ (take $u = p$). Thus the set of solutions for $b(p)$ and γ is given by $b(p) = -\omega$ and $\gamma \geq 0$, or $b(p) \geq -\omega$ and $\gamma = 0$.

Now a careful look of the reasoning above shows that it can be inverted: if $\check{R}'$, $\check{C}'$, b satisfy $(A7')$, $(A8')$, $(A10')$ and $\check{R}'(p) = 0$, then the associated R, C, b satisfy $(A7)$, $(A8)$, $(A10)$. This gives the first part of the theorem.

When $t \geq p$, we have:

$$\check{R}'(t) - \check{C}(t) = \inf\big(\inf_{p \leq u \leq t} \big[\check{R}(u) - \check{C}(u) + b(p)\big], \inf_{-\infty \leq u \leq p} \big[\check{R}(u) - \check{C}(u) + b(p)\big], \gamma\big).$$

We know that the last two terms are equal to $\check{R}'(p) = 0$. Thus

$$\check{R}'(t) - \check{C}(t) = \inf\big(\inf_{p \leq u \leq t} \big[\check{R}(u) - \check{C}(u) + B(p)\big], 0\big).$$

In the special case where we also assume that $b(p) = 0$, if $t \geq p$, we have:

$$\check{R}'(t) - \check{C}(t) = \inf\big(\inf_{p \leq u \leq t} \big[\check{R}(u) - \check{C}(u)\big], 0\big).$$

When $u = p$, the main term $\check{R}(p) - \check{C}(p)$ is equal to 0 by definition, so it covers the value 0 and we have:

$$\check{R}'(t) - \check{C}(t) = \inf_{p \leq u \leq t} \big[\check{R}(u) - \check{C}(u)\big].$$

Thus, for all $t \geq p$,

$$\check{R}'(t) = \inf_{p \leq u \leq t} \big[\check{R}(u) + \check{C}(t) - \check{C}(u)\big].$$

Both functions $\check{R}$ and $\check{C}$ are non-decreasing, equal to 0 at $t = p$, and non-negative over $[p, +\infty[$. This is exactly the dynamics of a Variable Capacity Node (with a initial point at $t = p$). $\square$

The RTC models combine upper and lower bounds on the traffic and on the services. Consider an RTC Greedy Processor as illustrated by Figure 3. Each traffic (or rather *event stream* in RTC) cumulative curve R can be constrained by a pair of *RTC arrival curves* $(\alpha^l, \alpha^u) \in \mathcal{F}_\uparrow \times \mathcal{F}_\uparrow$: for all $s \leq t$, let $R[s,t)$ the number of events that have arrived in the interval $[s,t[$, then

$$\alpha^l(t-s) \leq R[s,t) \leq \alpha^u(t-s).$$

In the same way, each resource cumulative curve C can be constrained by a pair of *RTC service curves* $(\beta^l, \beta^u) \in \mathcal{F}_\uparrow \times \mathcal{F}_\uparrow$: for all $s \leq t$, let $R[s,t)$ the number of processing cycles available in the interval $[s,t[$, then

$$\beta^l(t-s) \leq C[s,t) \leq \beta^u(t-s).$$

If inputs are shaped by such curves, one can propagate the same kind of constraints to the outputs. Such output constraints are given by the following formulas proved in [19]: with the notation of Figure 3,

$$\begin{aligned}
\alpha'^u &= \min\big((\alpha^u * \beta^u) \oslash \beta^l, \beta^u\big);\\
\alpha'^l &= \min\big((\alpha^l \oslash \beta^u) * \beta^l, \beta^l\big);\\
\beta'^u &= (\beta^u - \alpha^l)\overline{\oslash}0;\\
\beta'^l &= (\beta^l - \alpha^u)\overline{*}0 = (\beta^l - \alpha^u)_{\uparrow}.
\end{aligned}$$

The proofs in [19] (Appendix A, pages 201-204) use additional assumptions on the model: the existence of arbitrary small instants p such that $b(p) = 0$. As a consequence of Theorem 2, over such intervals $[p, +\infty[$, the RTC Greedy Processors behave exactly as NC Variable Capacity Nodes. Consequently one can retrieve results from Variable Capacity Nodes like the link with strict service curves of Theorem 1.

Corollary 1. *Consider an RTC Greedy Processor where inputs R and C are given, and C has a lower RTC service curve β^l. Consider a solution (R, C, R', C', b) satisfying Equations (A7), (A8), (A10). Suppose that $\exists p \in \mathbb{R}$ such that $B(p) = 0$. Then the trajectory $(\check{R}, \check{R}')$ (where $\check{F}(.) = F[p, .)$) admits β^l as a strict service curve.*

Remark 1 (Old RTC vs new RTC)**.** *The comparison above is drawn between Variable Capacity Nodes and a* recent *presentation of RTC. Note that in the very first papers about RTC the Greedy Processors were defined exactly as Variable Capacity Nodes [18, 17]. The reason for choosing a new RTC formalism (mainly with functions defined over all $\mathbb{R}$) remains unclear for us, although the information might be present in the literature.*

Remark 2 (What is a proof ? Equations vs interpretation)**.** *A temptation is to do proofs "by interpretation", i.e. one describes with words what is happening in the system. Ok it can be an useful approach and intuitions can be mentioned but the final proof should stick to the mathematical formalism, otherwise there are great risks of errors or ambiguities. Consequence: it requires very rigorous definitions at the beginning.*

4 Composition of service curves: convolution

We have now compare different notions of service curves under various versions of the Network Calculus framework. In the two next section, we study more precisely simple and strict service curves, which are widely used in the NC community. We focus on the stability of the type of curves with two operators: composition and remaining service curves. In this section, we focus on the composition of service curves (servers in tandem).

4.1 Stability/instability results

Theorem 3. *Consider two servers in tandem, of respective service curves β_1 for A and β_2 for B. Then*

1. *$\beta_1 * \beta_2$ is a service curve for A.*
2. *If β_1 and β_2 are strict, $\beta_1 * \beta_2$ is not necessarily strict.*

Proof. The first part of the theorem is a classical property of the concatenation of servers, which can be found in [12, 6].

We then only show an example to illustrate the fact that $\beta_1 * \beta_2$ is not necessarily a *strict* service curve for their concatenation. Here is a counterexample with burst-delay strict service curves. As a direct application of definitions, a node admits δ_T as a strict service curve if and only if the duration of any backlogged period is at most T. Consider the system on Figure 4 composed of two nodes in tandem. Suppose that Node 1 (resp. Node 2) works as the following server: as soon as a positive amount of data arrives, it waits a time $T_1 = 2$ (resp. $T_2 = 3$) while its queue fills up, and then it serves instantaneously all the data in its queuing. Node 1 (resp. Node 2) clearly has δ_{T_1} (resp. δ_{T_2}) as a strict service curve. Figure 4 shows what happens to a flow crossing Node 1 and 2 with the cumulative arrival function $A(t)$. The duration of the backlogged period is 9, and thus $\delta_{T_1} * \delta_{T_2} = \delta_{T_1+T_2}$ is not a strict service curve since $T_1 + T_2 = 5$.

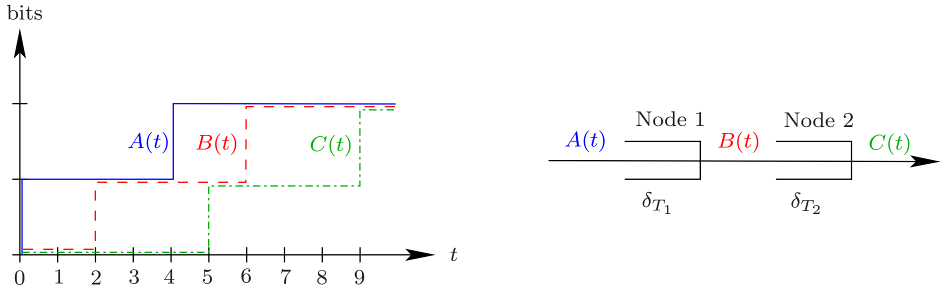


Figure 4: Tandem: strict service curve $*$ strict service curve $\neq$ strict service curve.

□

4.2 An intermediate notion of service curve?

One of our main motivation was to try to define an intermediate notion of service curve that would in particular be preserved with the concatenation. We now show that this is not possible: the next theorem shows that considering only the concatenation, the notion of intermediate service curve must coincide with that of service curve. As we will see later, no residual service curve notion can be defined from the notion of service curve. As a consequence, no notion of intermediate service curve can be preserved both through concatenation and "residuation".

Let us recall the properties that our intermediate service should satisfy to be preserved with concatenation (we denote by $\mathcal{S}_{inter}(\beta)$ the relation between trajectories that must be satisfied for an intermediate service curve β):

1. $\mathcal{S}_{strict}(\beta) \subseteq \mathcal{S}_{inter}(\beta) \subseteq \mathcal{S}_{simple}(\beta)$;
2. $\mathcal{S}_{inter}(\beta_2) \circ \mathcal{S}_{inter}(\beta_1) = \mathcal{S}_{inter}(\beta_1 * \beta_2)$: if β_1 and β_2 are the respective intermediate service curves for two servers S_1 and S_2 in tandem, then $\beta_1 * \beta_2$ is an intermediate service curve for the concatenation of those two servers.

Theorem 4. *No notion of intermediate service curve can be strictly more restrictive than the notion of service curve.*

Proof. The idea is two decompose one server with a (strict) service curve β into a sequence of servers with of service curves β_i such that $\beta = \ast \beta_i$, and to compare the departure processes when servers are used as strict service curves or as service curves. We will see that when the number of servers grows to infinity, the behaviors tend to be the same. Note that as the convolution is associative, the departure process corresponding to an arrival process A though a server offering a service curve β is the same as the departure process through servers in tandem offering β_i as service curves. As a consequence, we will mainly focus of the departure processes where the services are strict.

Pure delay curve

Before proving the result for any convex service curve, let first prove it for a curve δ_d, where $\delta_d : t \mapsto 0$ if $t \leq d$; $+\infty$ otherwise. This case gives the intuition that lays behind the proof.

Let A be a cumulative arrival process. Let n be the number of servers in tandem and d/n be the pure delay of each server (each server offers a service curve $\delta_{d/n}$). Let us compute the delay of each packet in the process. Let B_n be the departure process of A after crossing the n servers in tandem when the services are strict and exact and let t_0 be the arrival time of the packet we want to compute the delay.

The delay in the first server in upper bounded by d/n, and is served at time $t_1 \leq t_0 + d/n$. The process after the first server is bursty (it is a stair function), and time between to bursts is at least d/n. Then, at the second server, as every packet of the backlogged period has been served at the same time, the packet arrives at the beginning of a busy period and will be served at time $t_2 = t_1 + d/n$. It is obvious that the same will hold for the $n-2$ next servers. Then, the overall delay δ for this packet is bounded by:

$$\frac{n-1}{n} d \leq \delta \leq d.$$

When n grows to infinity, the delay of the packet tends to d. As this holds for every packet, $B_n(t) \to A(t-d)$.

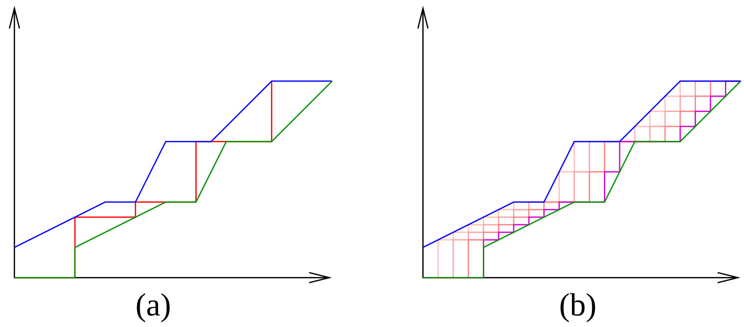


Figure 5: Example of output service curve after 4 servers in tandem. (a) the arrival process (blue) and departure processes for strict (red) and non strict (green) service curves. (b) Arrival (blue) and departure process for non strict service curve (green). From light to dark, the departure process after each server with strict service curve $\delta_{d/4}$.

Elementary segment service curve

We now prove the theorem for a curve of the following form: $\beta_{\lambda,d} : t \mapsto \lambda t$ if $t \leq d$; $+\infty$ otherwise. This is the result that will be use to prove the theorem for any convex curve.

First, remark that, as in the previous case, $\beta_{\lambda,d} = \beta^n_{\lambda,d/n}$. Let A be a cumulative arrival process. The same cumulative departure processes can be obtain if we consider that the flow crosses one server with minimal service curve $\beta_{\lambda,d}$ or n servers with minimal service curve $\beta_{\lambda,d/n}$ in tandem. The output process obtained when the service is not strict, if the servers are exact, is $A * \beta_{\lambda,d}$. Graphically, this is the minimal closure of the points of the segments of slope λ and of length d when the starting points of the segments follow A, as depicted in Figure 6.

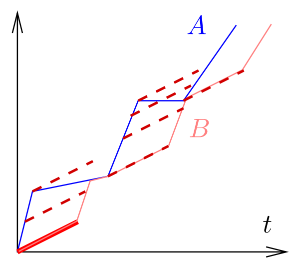


Figure 6: Convolution with an elementary segment: in bold and plain, $\beta_{\lambda,d}$, in bold and dotted, some segments following A, $B = A * \beta_{\lambda,d}$.

The behavior is different when the service curves are strict. Let us see what happen (we will suppose that the services are exact to construct the processes). The backlogged period of a process through a server of strict service curve $\beta_{\lambda,d/n}$ is at most d/n. If it is exactly d/n, then the slope of the departure process in that backlogged period is exactly λ. Otherwise, this means that the arrival process has intersects the segment of slope λ that starts at the beginning of the backlogged period, which means that the backlogged period ends at the intersection, and a new one may begin. The remaining of the segment of length d would not be of interest for the computation of the convolution, as it is covered by the segment that begins at the intersection. After crossing one such server, the slopes of B', the departure process after crossing the first server at most λ, and there may be discontinuities (bursts). When the delay is strictly less that λ, this means that the delay is 0. When there are discontinuities, the backlogged period ending with it is of length d/n. The different cases that can occur are depicted in Figure 7.

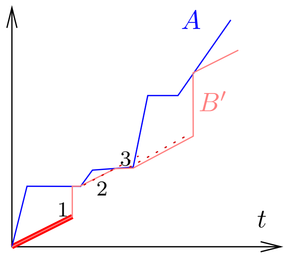


Figure 7: Different cases for elementary segment strict service curves. 1: the backlogged period is exactly d/n, the slope of the departure process during this period is λ and there is a burst at the end of the backlogged period. 2: the backlogged period is strictly less than d/n, as the segment of slope λ and of length d/n intersects A. After the intersection, the segment is above the curves or segments that begin the next backlogged period. 3: the slope of the arrival process is less than λ at the end of a backlogged period, then, the departure process is not delayed.

For what has been said before, one can deduce that the departure process $B^{(1)}$ can be written as

$$B^{(1)}(t) = \min_{s\in\mathcal{D}\cap[0,t]} A(s) + \beta_{\lambda,d/n}(t-s),$$

where $\mathcal{D} \subset \mathbb{R}_+$ such that $0 \in \mathcal{D}$ and $\forall s \in \mathcal{D}$, $\exists u \in \mathcal{D}$ such that $u - s \leq d/n$.

Consider $B^{(i)}$ the departures processes after crossing the i-th server ($B^{(i-1)}$ is the the arrival process in the i-th server). The same observations can be made than between A and $B^{(1)}$: the slopes of $B^{(i)}$ are at most λ, there can be bursts. Backlogged period (of positive length) necessarily start when there is a burst. For each burst in $B^{(i)}$, $i > 1$, say at time t, there was a burst at time $t - d/n$ for $B^{(i-1)}$, and by induction, at time $t - (i-1)d/n$ for $B^{(1)}$, and the beginning of the backlogged period in the first server is at time $t - id/n$. Bursts correspond to the end of backlogged period of length d/n. Now, if there is a backlogged period of length less that d/n in the i-th server that starts at time t, this means that $s \mapsto B^{(i-1)}(t) + \beta_{\lambda,d/n}(s)$ and $s \mapsto B^{(i-1)}(t+s)$ intersect each other (which corresponds to case 2 of Figure 7). The slope of $B^{(i-1)}$ at that intersection (t') is strictly less than λ, so $B^{(i-1)}(t') = A(t')$. The beginning of the corresponding backlogged period in the first server is $t - (i-1)d/n$. Between $t - (i-1)d/n$ and t', $B^{(i)}$ is affine of slope λ. Applying this to $i = n$, one gets, with $B_n = B^{(n)}$,

$$B_n(t) = \min_{s\in\mathcal{D}\cap[0,t]} A(s) + \beta_{\lambda,d}(t-s).$$

Elements in $\mathcal{D}$ correspond to the beginning of backlogged period in the first server, taking into account the periods of length 0. The horizontal distance between B and B_n is then bounded by d/n and then, when $n \to \infty$, $B_n \to B$.

Convex service curve

Let β be a convex service curve piece-wise affine, composed of k finite segments of respective slope

λ_i and length λ_i and of one semi-infinite segment of slope λ. we know that $\beta = \ast_{i=1}^{k} \beta_{\lambda_i, d_i} * \beta_{\lambda, \infty}$. One can decompose that service curve into $kn+1$ curves: $\beta_{\lambda,\infty}$, and n curves of each $\beta_{\lambda_i, d_i/n}$.

The behavior of a server with service curve $\beta_{\lambda,\infty}$ is the same if the service is strict of not when the service is exact. So put this server first. The order of the servers is depicted on Figure 8. Let A be an arrival process in the concatenation of servers. We denote by B_0 (resp. B'_0) the departure process after the first server when the service is simple (resp. strict), B_i (resp. B'_i) the departure process after n the servers $\beta_{\lambda_i, d_i/n}$ when the service is simple (resp. strict). One has $B_0 = B'_0$.

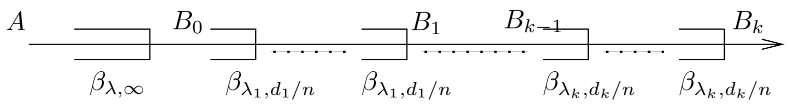


Figure 8: Decomposition of a server with convex service curve into a concatenation of elementary servers.

We now prove by induction on k that $B'_k \geq B_k \geq B'_k(. - \sum_{i=1}^{k} d_i/n)$. The case $k = 1$ corresponds to the case of an elementary segment of the previous paragraph. Suppose that the result is true for k elementary segments. We prove it for $k+1$ segments using the monotony of the convolution: if $f_1 \geq f_2$, then $f_1 * g \geq f_2 * g$ and the previous paragraph to the shifted process $B'_{k+1}(. - \sum_{i=1}^{k} d_i/n)$

When the services are strict, the departure process is B'_{k+1} when the arrival process in the servers $\beta_{\lambda_{k+1}, d_{k+1}/m}$ is B'_k and is $B'_{k+1}(. - \sum_{i=1}^{k} d_i/n)$ when the arrival process in the servers $\beta_{\lambda_{k+1}, d_{k+1}/m}(. - \sum_{i=1}^{k} d_i/n)$. When the services are not strict, the departure process is

$$\begin{aligned} B_k * \beta_{\lambda_{k+1}, d_{k+1}} &\geq B'_k(. - \sum_{i=1}^{k} d_i/n) * \beta_{\lambda_{k+1}, d_{k+1}} \\ &\geq B'_{k+1}(. - \sum_{i=1}^{k+1} d_i/n). \end{aligned}$$

When n grows to infinity, this horizontal distance between B_k and B'_k tends to 0. This finishes the proof.

□

5 Residual service curves

This section is dedicated to the second aspect of NC we want to study: the remaining service curves: what service can be guaranteed for a flow if it suffers from another flow in the same server? As we have to deal with several flows, the use of arrival curves will be necessary here. Moreover, we will study here two type of service policy: blind multiplexing (any service policy can be used by the server) and the fix priority (FP) policy (some flows have higher priority than others). Remaining service curves under bling multiplexing has been extensively studied in the last years ([]), whereas fix priority policy is rather used with the RTC model ([]). We show that we can translate the results from RTC (hence from VCN service curves) with strict service curves.

We will use the following notation for priorities: if a flow F_2 has higher priority tha F_2, we write $F_1 \succ F_2$.

5.1 Stability results for a fluid model

Theorem 5. *Consider a server offering a strict service curve β to $A_1 + A_2$ and suppose that flow A_1 is α_1 upper-constrained.*

1. *The server service a service curve $(\beta - \alpha_1)_\uparrow$ for A_2.*

2. *If $A_1 \succ A_2$, then $(\beta - \alpha_1)_\uparrow$ is a strict service curve.*

As far a strict service curves are concerned, a similar result is already known for the RTC model with fix priorities. Nevertheless, we will give a direct proof in the Network Calculus framework. In this latter framework, the existing result is that $(\beta - \alpha_1)_+$ is a service curve. Having the positive, non-decreasing upper closure can be useful when dealing with sub-additive, non-concave arrival curves.

Proof. In addition to the two items of the theorem, we will give an example to illustrate that it there is no fix priority among the two flows, the remaining service curve may not be strict.

We first show that if A_1 has a higher priority than A_2, then the server offers a strict service curve $(\beta - \alpha_1)_\uparrow$ to A_2.

Let u and v, $u < v$ be times in the same backlogged period for A_2 in the server. As a consequence, u and v are in the same backlogged period of the aggregate flow, and we have:

$$B_1(v) + B_2(v) \geq B_1(u) + B_2(u) + \beta(v-u).$$

Let p be the start of the backlogged period of u for F_1. As the period between p and u is a backlogged period for F_1, and u is in a backlogged period for F_2, no data can be served for F_2 between p and v and we have:

$$\begin{array}{rcl} B_1(u) - B_1(p) & \geq & \beta(u-p) \\ B_2(u) - B_2(p) & = & 0 \\ A_1(p) - B_1(p) & = & 0. \end{array}$$

Between times p and v, we also have:

$$B_2(v) - B_2(p) + B_1(v) - A_1(p) \geq \beta(v-p).$$

As $B_1(v) - A_1(p) \leq A_1(v) - A_1(p) \leq \alpha_1(v-p)$, we obtain:

$$B_2(v) - B_2(p) \geq \beta(v-p) - \alpha_1(v-p).$$

One can apply the formula above for every $w \in [p, v]$, w remains in the same backlogged period as v:

$$\forall w \in [p, v] \;\; B_2(w) - B_2(p) \geq \beta(w-p) - \alpha_1(w-p).$$

As B_2 is non-decreasing, we also have

$$\forall w \in [p, v] \;\; B_2(v) - B_2(p) \geq B_2(w) - B_2(p) \geq \beta(w-p) - \alpha_1(w-p)$$

and

$$\begin{array}{rcll} B_2(v) - B_2(p) & \geq & \sup_{w \in [p,v]} \beta(w-p) - \alpha_1(w-p) & (1) \\ & \geq & \sup_{s \in [0,v-p]} \beta(s) - \alpha_1(s) & (2) \\ & \geq & \sup_{s \in [0,v-u]} \beta(s) - \alpha_1(s). & (3) \end{array}$$

As $B_2(u) = B_2(p)$, we have

$$B_2(v) - B_2(u) \geq \sup_{s \in [0,v-u]} \beta(s) - \alpha_1(s).$$

Moreover, as B_2 is non-decreasing,

$$B_2(v) - B_2(u) \geq \sup_{s \in [0,v-u]} (\beta(s) - \alpha_1(s))_+.$$

To show that under blind multiplexing, $(\beta - \alpha_1)_\uparrow$ is a service curve is quite straightforward from what has been written above: Equations (1) and (2) still hold, and $A_2(p) = B_2(p)$, so

$$B_2(v) \geq A_2(p) + \sup_{s \in [0, v-p]} \beta(s) - \alpha_1(s) \geq A_2 * (\beta - \alpha_1)_\uparrow.$$

We now give an example where the remaining service curve is not strict. Suppose for instance that we are in the infinitesimal data model and consider a node which serves data at a constant rate 2. Such a node has $\beta(t) = 2t$ as a strict service curve. Let two flows cross this node with respective cumulative arrival functions $A_1(t) = t + 1$ and $A_2(t) = t$. An arrival curve of Flow 2 is $\alpha_2(t) = t$. Figure 9 shows the trajectory of the system for the following policy: at first, for $0 \leq t \leq 1$, Flow 1 is given top priority, then for $t > 1$, Flow 2 is given top priority. Since the sum $A_1(t) + A_2(t) = 2t + 1$, the node is always backlogged and the sum of the cumulative departure functions is $B_1(t) + B_2(t) = 2t$. It can be checked that $\beta_1(t) = (\beta - \alpha_2)_+(t) = t$ is a service curve for Flow 1 ($B_1 \geq A_1 * \beta$) but it is not a *strict* service curve: during the period $1 \leq t \leq 2$, Flow 1 has some data backlogged in the node but this data is not served at all.

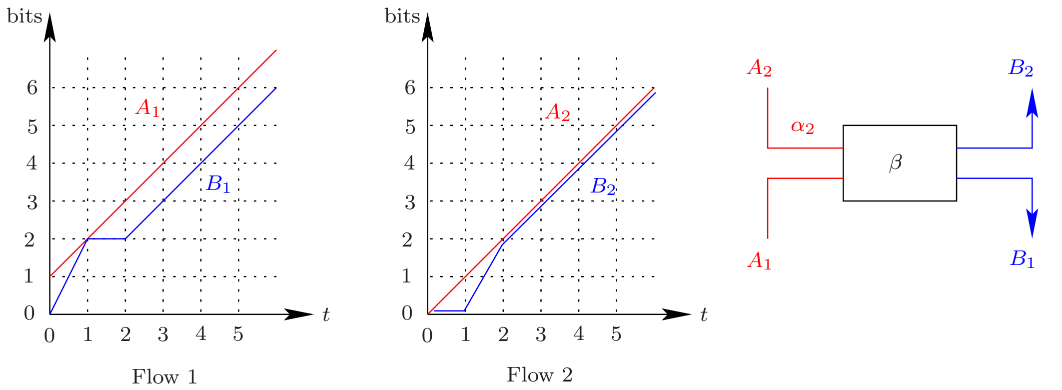


Figure 9: Residual service curves are not necessarily strict.

□

5.2 What about the packet size?

Theorem 6. *Consider a server with a strict service curve β for $A_1 + A_2$. Suppose that flow A_1 is α_1 upper-constrained, that $A_1 \succ A_2$ and that $\ell_{2,\max}$ is the maximum size of a packet in F_2.*

1. *The server guaranties a service curve $(\beta - \alpha_1)_\uparrow$ for A_2.*
2. *The server guaranties a strict service curve $(\beta - \alpha_1 - \ell_{2,\max})_\uparrow$ for A_2.*
3. *The server guaranties a strict service curve $(\beta - \ell_{2,max})_\uparrow$ for A_1.*

Proof. The first part of the theorem is a direct consequence of Theorem 5.1. The second and third parts can be proved the same way as Theorem 5 or one can make the following observations to apply that Theorem 5.2: as the server is non-preemptive, when a packet of F_2 is served, a packet of flow F_1 can arrive in the system, starting a backlogged period for this server. But the packet of F_2 being served by the system can be seen as a priority flow. This flow is upper-constrained by $t \mapsto \ell_{2,\max}$, hence $(\beta - \ell_{2,max})_\uparrow$ is a strict service curve for A_1.

The same way, in the proof of Theorem 5, A_1 can be replaced by $A_1 + \ell_{2,max}$, a flow upper-constrained by $t \mapsto \ell_{2,\max} + \alpha_1(t)$, hence $(\beta_1 - \alpha_1 - \ell_{2,\max})_\uparrow$ in a strict service curve for A_2.

We can also adapt the example in proof of Theorem 5 to show that $(\beta - \alpha_1)_\uparrow$ may not be strict. We still take $\beta(t) = 2t$, $A_2(t) = \alpha_2(t) = 1 + t$. We set $\ell_{2,\max} = 1$, $\alpha_1(t) = 1/2 + t$ and $A_1(t + 1/2) = \alpha_1(t)$ or 0 if $t < 1/2$. At time 0, a packet of size $\ell_{2,\max}$ arrives, so it is served as there is no packet of flow 1 in the server at time 0. At time 1, the packet of size $\ell_{2,\max}$ is served, so that packets of F_1 can be served. After that, the behavior is the same as in the proof of Theorem 5: between time 1 and 2, no packet of flow 2 is served, whereas $(\beta - \alpha_1)_{\uparrow}(t) = 2(t - 1/4)_+$, and the interval during which no packet of F_2 is served in a backlogged period for tat flow cannot exceed 1/4.

□

As a direct application, this theorem can be used to compute a minimal (strict) service curve for each flow of a server, with fix priorities on the flows.

Corollary 2. *Consider a single server with a minimal strict service curve β and m flows crossing it, $F_1, \ldots, F_m$. The maximum packet size of flow F_i is $\ell_{1,\max}$ and is upper-constrained by the arrival curve α_i. Flows are ordered in their order of priority : $F_i \prec F_j \Leftrightarrow i < j$. For each $i \in \{2, \ldots, m\}$,*

1. *$(\beta - \sum_{j<i} \alpha_j - \vee_{k>i} \ell_{k,max})_+$ is a service curve for flow F_i.*
2. *$(\beta - \sum_{j<i} \alpha_j - \vee_{k \geq i} \ell_{k,max})_+$ is a strict service curve for flow F_i.*

5.3 Composition of residual services: nested flows with fix priorities

In this section, we study more precisely scenarii where the strict service curve composition would have been useful. We show that even if the strict property is lost, computations are still possible and valid. There is one kind of scenarii where direct composition is efficient: when we have servers in tandem with nested flows.

We will use the following notation:

- F_i represent flows;
- If there are n servers in tandem, $F_i^{(j)}$, represent the arrival process of flow i in the $j+1$-th server and the departure process of the j-th server;
- $F_i^{(0)}$ is α_i-upper constrained;
- β_j is the strict service curve of the j-th server.

Consider n servers in tandem aditting respective strict service curves β_j. The service curve of one flow is $\beta_1 * \cdots * \beta_n$.

Lemma 4 (One nested flow)**.** *Consider a flow F_2 crossing the servers $\beta_b, \ldots, \beta_e$ that is α_2-upper constrained. A service curve for flow F_1 is $\beta_1 * \cdots \beta_{b-1} * (\beta_b * \cdots * \beta_e - \alpha_2)_+ * \beta_{e+1} * \cdots * \beta_n$.*

Proof. First consider the service curve between servers b and server e, and then the other are obtained by concatenation, which is valid for any service curve.

$\forall t_i \in \mathbb{R}_+$, there exists t_{i-1}, first instant of backlogged period such that

$$F_1^{(i)}(t_i) + F_2^{(i)}(t_i) \geq F_1^{(i-1)}(t_{i-1}) + F_2^{(i-1)}(t_{i-1}) + \beta_i(t_i - t_{i-1}).$$

Then, $\forall t_e \in \mathbb{R}_+$, there exists $t_{b-1}, \ldots, t_{e-1}$ such that

$$F_1^{(i)}(t_e) + F_2^{(i)}(t_e) \geq F_1^{(b-1)}(t_{b-1}) + F_2^{(b-1)}(t_{b-1}) + \sum_{i=b}^{e} \beta_i(t_i - t_{i-1}),$$

and

$$F_1^{(e)}(t_e) \geq F_1^{(b-1)}(t_{b-1}) + \sum_{i=b}^{e} \beta_i(t_i - t_{i-1}) - \alpha_2(t_e - t_{b-1}).$$

Moreover, $F_1^{(e)}(t_e) \geq F_1^{(b-1)}(t_{b-1})$, so

$$F_1^{(e)}(t_e) \geq F_1^{(b-1)}(t_{b-1}) + (\ast_{i=b}^{e} \beta_i(t_e - t_{b-1}) - \alpha_2(t_e - t_{b-1}))_+.$$

□

Suppose now that there exist m transversal flows $F_1, \ldots, F_m$. Flow F_i interfere with a flow F_0 on line on the n servers previously defined on the interval $I_i = \{b_i, \ldots, e_i\}$ and we suppose that the flows are nested: $\forall i, j$, either $I_i \cap I_j = \emptyset$ or $I_i \subseteq I_j$ or $I_j \subseteq I_i$. Furthermore, suppose that the priorities of the flows is fixed, forms a total order, and respects the following rule:

$$(R) \qquad \forall i, j \in \{1, \ldots, m\}, \qquad F_i \succ F_j \Rightarrow I_i \subseteq I_j.$$

For $i \in \{0, \ldots, m\}$, we denote by P_i the set of flows that has higher priority than F_i and interfere with it ($P_i = \{j \in \{0, \ldots, m\} \mid F_j \succ F_i$ and $I_j \subseteq I_i\}$ and by P_i^{im} the set of flows of minimal priority among those that have higher priority than F_i: $P_i^{im} = \{j \in P_i \mid \forall k \in P_i \setminus \{j\}\ F_k \succ F_j\}$. Finally we denote by N_i the set of servers crossed by F_i that are not crossed flows of higher priority: $N_i = \{j \in I_i \mid \forall k \in \{1, \ldots, m\},\ F_k \succ F_i \Rightarrow j \notin I_k\}$.

We are going to prove by induction that the service can be removed by one by one in decreasing order of priorities to obtain a global remaining service curve, although the strictness property of the service curves in not preserved. More formally, we have the following theorem:

Theorem 7. *If flows respect rule (R) and using the notations defined above, for every $i \in \{0, \ldots, m\}$, β'_i is a service curve for flow F_i.*

$$\beta'_i = \mathop{\ast}_{j \in N_i} \beta_j * \mathop{\ast}_{k \in P_i^{im}} (\beta'_k - \alpha_k)_+.$$

Proof. We show this result by induction. Our induction hypothesis (H_m) for m interfering flows is: consider m flows $F_1, \ldots, F_m$ interfering with a line of n servers with fixed priorities respecting rule (R). For any aggregate flow F_{m+1} with lower priority than $F_1, \ldots, F_m$,

1. For each server $j \notin \bigcup_{i \in \{1, \ldots, m\}} I_i$, $\forall t \in \mathbb{R}_+$, $\exists u_j$ such that
$$F_{m+1}^{(j)}(t) \geq F_{m+1}^{(j-1)}(u_j) + \beta_j(t - u_j).$$
2. For each $k \in P_{m+1}^{im}$, $\forall t \in \mathbb{R}_+$, $\exists u_k$ such that
$$F_{m+1}^{(e_k)}(t) \geq F_{m+1}^{(b_k-1)}(u_k) + (\beta'_k - \alpha_k)_+(t - u_k).$$

(H_0) holds: as there is no interfering flows, only the first case has to be considered, for any server. By definition of strict service curves, the first case holds for any server.

Suppose that (H_m) holds. We now show that (H_{m+1}) also holds. Consider a line of n servers with $m + 1$ flows with fixed priorities. Without loss of generality, one can suppose that F_{m+1} has the lowest priority. So, (H_m) holds for $F_1, \ldots, F_m$. Also consider F_{m+2} a flow with lower priority than F_{m+1}. Three cases can occur:

1. Server j does not interfere with $F_1, \ldots, F_{m+1}$. Then as server j offers a strict service curve β_j, $\forall t \in \mathbb{R}_+$, $\exists u_j$ such that
$$F_{m+2}^{(j)}(t) \geq F_{m+2}^{(j-1)}(u_j) + \beta_j(t - u_j).$$
2. For each $k \in P_{m+2}^{im}$ such that $I_k \cap I_{m+1} = \emptyset$, as F_k has higher priority that F_{m+1}, those two flow do not interfere with each other. Then, using the induction hypothesis, we have $\forall t \in \mathbb{R}_+$, $\exists u_k$ such that
$$F_{m+2}^{(e_k)}(t) \geq F_{m+2}^{(b_k-1)}(u_k) + (\beta'_k - \alpha_k)_+(t - u_k).$$
3. Flows F_{m+2} and F_{m+1} interfere on I_{m+1}. Every other flow that interfere with F_{m+1} has higher priority and respects rule (R). Let $\tilde{\beta}_1, \ldots, \tilde{\beta}_\ell$ the ordered sequence of service curves crossed by F_{m+2} and F_{m+1} (i.e. only those crossed by F_{m+1}), $\tilde{\beta}_q$ being β_j for some j or

$(\beta'_k - \alpha_k)_+$ for some k satisfying (H_m). For every u_ℓ, there exists $u_0, \ldots, u_{\ell-1}$ such that for each $p \in \{1, \ldots, \ell\}$,

$$F_{m+2}^{(r_p)}(u_p) + F_{m+1}^{r_p}(u_p) \geq F_{m+2}^{q_p}(u_{p-1}) + F_{m+1}^{q_p}(u_{p-1}) + \tilde{\beta}_p(u_p - u_{p-1}),$$

with $r_p = j$ and $q_p = j-1$ if $\tilde{\beta}_p = \beta_j$ or $r_p = e_k$ and $q_p = b_k - 1$ if $\tilde{\beta}_p = (\beta'_k - \alpha_k)_+$. Moreover, as the servers are in sequence, we have $r_p = q_{p+1}$ for $p \in \{1, \ldots, \ell - 1\}$ and $q_1 = b_{m+1} - 1$ and $r_\ell = e_{m+1}$.

Let add those ℓ inequations, we get:

$$F_{m+2}^{(e_{m+1})}(u_\ell) + F_{m+1}^{(e_{m+1})}(u_\ell) \geq F_{m+2}^{(b_{m+1}-1)}(u_0) + F_{m+1}^{(b_{m+1}-1)}(u_0) + \sum_{p=1}^{\ell} \tilde{\beta}_p(u_p - u_{p-1}).$$

As $F_{m+1}^{(e_{m+1})}(u_\ell) - F_{m+1}^{(b_{m+1}-1)}(u_0) \leq F_{m+1}^{(b_{m+1}-1)}(u_\ell) - F_{m+1}^{(b_{m+1}-1)}(u_0) \leq \alpha_{m+1}(u_p - u_0)$ and $F_{m+2}^{(e_{m+1})}(u_\ell) \geq F_{m+2}^{(b_{m+1}-1)}(u_0)$, we have

$$F_{m+2}^{(e_{m+1})}(u_\ell) \geq F_{m+2}^{(b_{m+1}-1)}(u_0) + (\sum_{p=1}^{\ell} \tilde{\beta}_p(u_p - u_{p-1}) - \alpha_{m+1}(u_p - u_0))_+$$

and

$$F_{m+2}^{(e_{m+1})}(u_\ell) \geq F_{m+2}^{(b_{m+1}-1)} * (\ast_{p=1}^{\ell} \tilde{\beta}_p - \alpha_{m+1})_+$$

(H_{m+1}) is then satisfied. The rest of the proof is straightforward.

□

6 Conclusion

This report has two aims: comparing several models having their root in Network calculus and comparing different notions of service curves in the original Network calculus model.

First, we show that the main alternative to Network calculus (NC), real-time calculus (RTC), is in fact very similar to NC. The service curve in RTC corresponds to the strict service curve in NC.

Second, we compare strict and simple service curves, and explicit what can be done or cannot be done with which type of service curve, concerning the composition of service curves and the computation of residual service curves. Moreover, we explain why we cannot define a new notion of service curve which enables and is stable with those two operations.

Although looking a bit formal, this work is relevant, as it showed some important details we - and, to our knowledge, the NC community - were not conscious of. For example, as far as residual service curves are concerned, there is a difference between blind multiplexing and fixed priorities, event if the worst departure process for blind multiplexing is obtained with fixed priorities. Studying RTC helped us with such remarks.

This study is somewhat incomplete when dealing with the hierarchy between the types of service curves. Of course, the hierarchy has been recalled, but no precise new result has been given. For example, it has not been studied precisely the family of functions for which two kinds of service curves coincide, nor had been investigated if a (family of) service curve of one type can be modeled by a (family of) service curves of another type. This would be interesting for modeling purposes to know for example what kind of systems can be modeled using simple service curve, strict service curves or both. This will be the object of further work concerning service curves in Network Calculus.

Another point that has not been discussed here is the tightness issue, that is, whether, for a given server, there exists a trajectory whose maximum delay reaches the delay computed with NC methods. For example for the computations in the last section, it is now well-known that bounds are not tight is the general case, but it would be interesting to quantify that non-tightness and to characterize the cases of tightness. This formula should also be compared to the computations in RTC.

7 Acknowledgements

This work has been partially supported by the ONERA under the contract number CNRS 038361.

References

[1] F. Baccelli, G. Cohen, G.Y. Olsder, and J.P. Quadrat. *Synchronization and linearity*. Wiley, 1992.

[2] A. Bouillard, B. Cottenceau, B. Gaujal, L. Hardouin, S. Lagrange, M. Lhommeau, and E. Thierry. COmputational Issues in Network Calculus (COINC).

[3] A. Bouillard, B. Gaujal, S. Lagrange, and E. Thierry. Optimal routing for end-to-end guarantees: the price of multiplexing. In *Proceedings of Valuetools'07*, 2007.

[4] A. Bouillard, B. Gaujal, S. Lagrange, and E. Thierry. Optimal routing for end-to-end guarantees using network calculus. *Performance Evaluation*, 65(11-12):883–906, 2008.

[5] S. Chakraborty, S. Künzli, L. Thiele, A. Herkersdorf, and P. Sagmeister. Performance evaluation of network processor architectures: Combining simulation with analytical estimation. *Computer Networks*, 41(5):641–665, 2003.

[6] C. S. Chang. *Performance Guarantees in Communication Networks*. TNCS, Springer-Verlag, 2000.

[7] F. Ciucu, A.Burchard, and J. Liebeherr. A network service curve approach for the stochastic analysis of networks. In *Proceedings of SIGMETRICS'2005*, 2005.

[8] M. Fidler. A network calculus approach to probabilistic quality of service analysis of fading channels. In *Proceedings of GLOBECOM'2006*, 2006.

[9] N. Gollan, F. A. Zdarsky, I. Martinovic, and J. B. Schmitt. The disco network calculator. In *Proceedings of MMB'2008*, 2008.

[10] Y. Jiang. A basic stochastic network calculus. In *Proceedings of ACM SIGCOMM'2006*, 2006.

[11] Y. Jiang and Y. Liu. *Stochastic Network Calculus*. Springer, 2008.

[12] J.-Y. Le Boudec and P. Thiran. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, volume LNCS 2050. Springer-Verlag, revised version 4, may 10, 2004 edition, 2001.

[13] H. Schioler, J. J. Jessen, J. Dalsgaard, and K. G. Larsen. Network calculus for real time analysis of embedded systems with cyclic task dependencies. In *Proceedings of Computers and Their Applications*, 2005.

[14] H. Schioler, H.-P. Schwefel, and M. B. Hansen. Cync: a matlab/simulink toolbox for network calculus. In *Proceedings of Valuetools'2007*, 2007.

[15] J. B. Schmitt and F. A. Zdarsky. The disco network calculator: a toolbox for worst case analysis. In *Proceedings of Valuetools'2006*, 2006.

[16] J. B. Schmitt, F. A. Zdarsky, and I. Martinovic. Performance bounds in feed-forward networks under blind multiplexing. Technical Report 349/06, University of Kaiserslautern, Germany, 2006.

[17] L. Thiele, S. Chakraborty, M. Gries, A. Maxiaguine, and J. Greutert. Embedded software in network processors models and algorithms. In *Proceedings of EMSOFT'2001*, 2001.

[18] L. Thiele, S. Chakraborty, and M. Naedele. Real-time calculus for scheduling hard real-time systems. In *Proceedings of ISCAS'2000*, 2000.

[19] E. Wandeler. *Modular Performance Analysis and Interface-Based Design for Embedded Real-Time Systems*. PhD thesis, ETH Zurich, September 2006.

[20] E. Wandeler, L. Thiele, M. Verhoef, and P. Lieverse. System architecture evaluation using modular performance analysis - a case study. *Software Tools for Technology Transfer*, 8(6):649–667, 2006.

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399